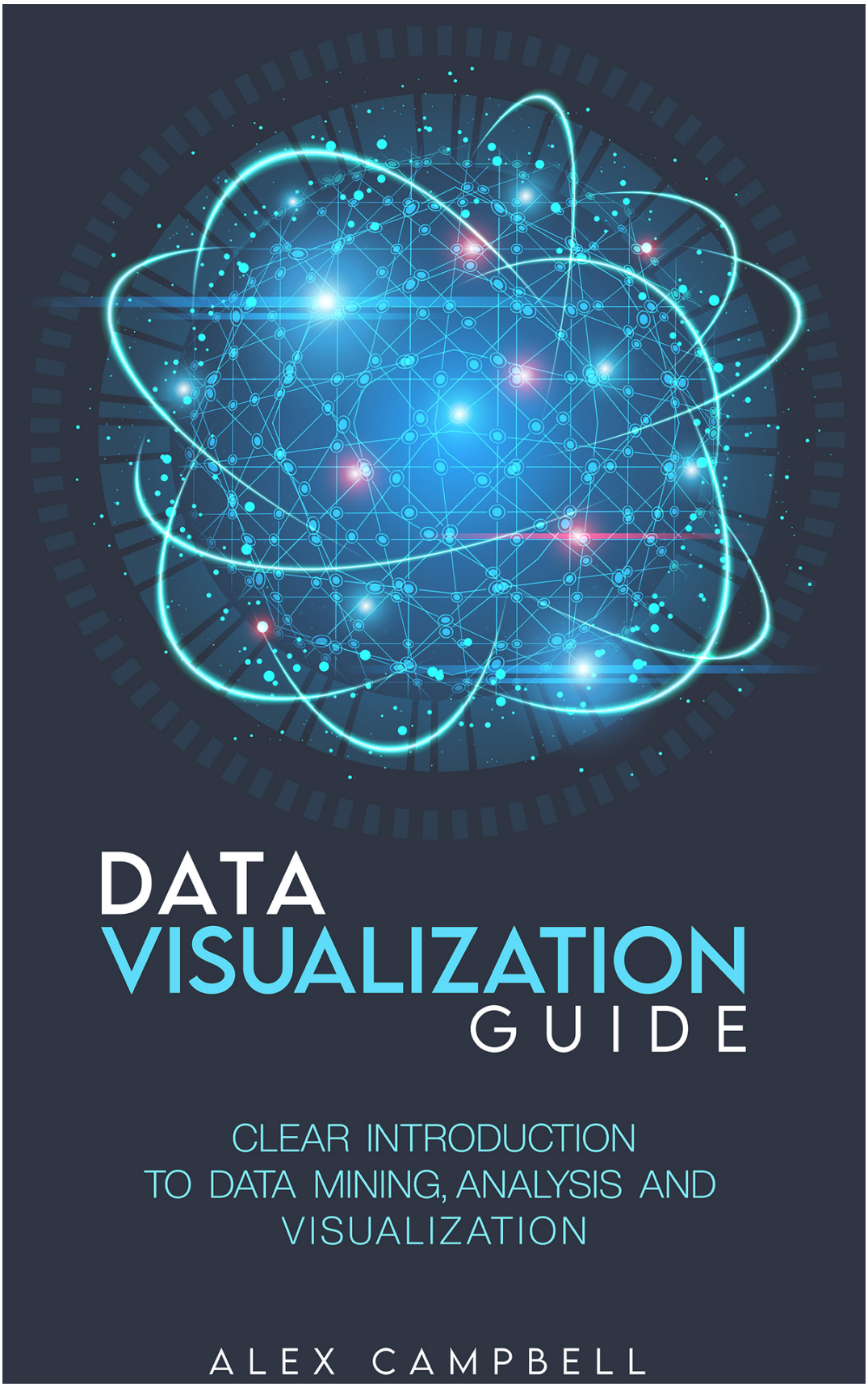
DATA VISUALIZATION GUIDE

CLEAR INTRODUCTION TO DATA MINING, ANALYSIS AND VISUALIZATION

ALEX CAMPBELL



DATA VISUALIZATION GUIDE

CLEAR INTRODUCTION TO DATA MINING, ANALYSIS AND VISUALIZATION

ALEX CAMPBELL

Book Description

Have you ever wondered how you can work with large volumes of data sets? Do you ever think about how you can use these data sets to identify hidden patterns and make an informed decision? Do you know where you can collect this information? Have you ever questioned what you can do with incomplete or incorrect data sets? If you said yes to any of these questions, then you have come to the right place.

Most businesses collect information from various sources. This information can be in different formats and needs to be collected, processed, and improved upon if you want to interpret it. You can use various data mining tools to source the information from different places. These tools can also help with the cleaning and processing techniques.

You can use this information to make informed decisions and improve the efficiency and methods in your business. Every business needs to find a way to interpret and analyze large data sets. To do this, you will need to learn more about the different libraries and functions used to improve data sets. Since most data professionals use Python as the base programming language to develop models, this book uses some common libraries and functions from Python to give you a brief introduction to the language.

If you are a budding analyst or want to freshen up on your concepts, this book is for you. It has all the basic information you need to help you become a data analyst or scientist.

In this book, you will:

- Learn what data mining is, and how you can apply in different fields.
- Discover the different components in data mining architecture.
- Investigate the different tools used for data mining.
- Uncover what data analysis is and why it’s important.
- Understand how to prepare for data analysis.
- Visualize the data.
- And so much more!

So, what are you waiting for? Grab a copy of this book now.

Data Visualization Guide

Clear Introduction to Data Mining, Analysis, and Visualization

© Copyright 2021 - All rights reserved. Alex Campbell.

The contents of this book may not be reproduced, duplicated or transmitted without direct written permission from the author.

Under no circumstances will any legal responsibility or blame be held against the publisher for any reparation, damages, or monetary loss due to the information herein, either directly or indirectly.

Legal Notice:

This book is copyright protected. This is only for personal use. You cannot amend, distribute, sell, use, quote or paraphrase any part or the content within this book without the consent of the author.

Disclaimer Notice:

Please note the information contained within this document is for educational and entertainment purposes only. Every attempt has been made to provide accurate, up to date and reliable complete information. No warranties of any kind are expressed or implied. Readers acknowledge that the author is not engaging in the rendering of legal, financial, medical or professional advice. The content of this book has been derived from various sources. Please consult a licensed professional before attempting any techniques outlined in this book.

By reading this document, the reader agrees that under no circumstances is the author responsible for any losses, direct or indirect, which are incurred as a result of the use of information contained within this document, including, but not limited to, —errors, omissions, or inaccuracies.

Table of Contents

Introduction

Chapter One: Introduction to Data Mining

Data Types Used in Mining

Pros and Cons of Data Mining

Applications of Data Mining

Challenges

Chapter Two: Data Mining Architecture

Data Source

Different Processes

Data Warehouse Server or Database

Data Mining Engine

Pattern Evaluation Module

Graphical User Interface (GUI)

Knowledge Base

Chapter Three: Data Mining Techniques

Classification

Clustering

Regression

Association Rules

Outer Detection

Sequential Patterns

Prediction

Chapter Four: Data Mining Tools

Orange Data Mining

SAS Data Mining

DataMelt Data Mining

Rattle

Rapid Miner

Chapter Five: Introduction to Data Analysis

Why Use Data Analysis?

Data Analysis Tools

Data Analysis Types

Data Analysis Process

Chapter Six: Manipulation of Data in Python

NumPy

Let's Start with Pandas

Chapter Seven: Exploring the Data Set

Chapter Eight: How to Summarize Data with Python

Obtain Information About the Data Set

Chapter Nine: Steps to Build Data Analysis Models in Python

Chapter Ten: How to Build the Model

Chapter Eleven: Data Visualization

Know Your Audience

Set Your Goals

Choosing the Right Charts to Represent Data

Conclusion

References

Introduction

Most organizations and businesses collect large volumes of data from various sectors and departments. This data is often unformatted, so you will need to find a way to process and clean it. Businesses can then use this information to make informed business decisions. They use data analysis and mining to interpret the data and collect the necessary information from the data set. These processes play an important role in any business. You can also use this type of analysis in your personal life. Data mining and analysis can be used to help you save money. Only when businesses know how to work with data can they know where they should reinvest the money and increase their revenue.

If you are new to the world of data, this book can be your guide. You can use the information to help you learn the basics of data mining and analysis. The book will also shed some light on the processes you can use to clean the data set, various processes and techniques you can use to mine and analyze information, and it will explain to you how you can visualize the data and why it’s important to represent data using graphs and other visuals.

Within these pages you will find information about the different techniques and algorithms used in data analysis, as well as provide you with different libraries you can use to manipulate and clean data sets. Most data analysis and mining algorithms are built using Python, and thus we will use the libraries and functions from Python in the book. You will also find a section including information about the process used to develop a model.

Before you work on developing different analysis techniques, you need to make sure you have the business problem or query in mind. It is important to bear in mind that any analysis you perform should be based on a business question. You need to make sure there is a foundation upon which you develop the model. Otherwise, the effort you put in will be unusable. Make sure you have all the details about why you are developing a model or collecting information before you put in the effort.

Chapter One: Introduction to Data Mining

I am sure you may have heard many people talk about data mining and how essential it is. But what is data mining? As the name suggests, data mining is the process of identifying and extracting hidden patterns, variables, and trends within any data set collected for your analysis. In simple words, the process of looking at data to identify any hidden patterns and trends of information that can be used to categorize the data into useful analysis is termed data mining or knowledge discovery of data (KDD). You can use data mining to convert raw data or information into data, which businesses can use.

It is important to remember that organizations often collect and assemble data from data warehouses. They use different data mining algorithms and efficient analysis algorithms to make informed decisions about their business. Through data mining, businesses can go through large volumes of data to identify patterns and trends, which would not be possible through simple analysis algorithms. We use complex statistical and mathematical algorithms to evaluate data segments and calculate a future event's probability. Organizations use data mining to extract the required information from large databases or sets to answer different business questions or problems.

Data mining and science are similar to each other, and in specific situations, these processes are carried out by one individual. There is always an objective for these processes to be performed. Data science and data mining processes include web mining, text mining, video and audio mining, social media mining, and pictorial data mining. This can be done with ease through different software.

Companies should outsource data mining processes since they have a lower operation cost. Some firms also use technology to collect various forms of data that cannot be located manually. You can find large volumes of data on different platforms, but there is very little knowledge that can be accessed from this data.

Every organization finds it difficult to analyze the various information collected to extract the information needed to solve any problem or make informed business decisions. There are numerous techniques and

instruments available to mine information from various sources to obtain necessary insights.

Data Types Used in Mining

You can perform data mining on the following data types:

Relational Databases

Every database is organized in the form of records, tables, and columns. A relational database is one where you can collect information from different tables or data sets and organize it in the form of columns, records, and tables. You can access this data easily without having to worry about individual data sets. The information is conveyed and shared through tables that increase the ease of organization, reporting, and searchability.

Data Warehouses

Every business collects data from different sources to obtain information to help them make well-informed decisions. They can do this easily using the process of data warehousing. These large volumes of data come from different sources, such as finance and marketing. The extracted information is then used for the purpose of analysis, which helps businesses make the right organizational decisions. A data warehouse has been designed to analyze data and not to process transactions.

Data Repositories

A data repository refers to the location where the organization can store data. Most IT professionals use this term to refer to the setup of data and its collection in the firm. For instance, they term a group of databases where different kinds of data are stored.

Object-Relational Database

An object-relational database is a combination of a relational database and an object-oriented database model. This database uses inheritance, classes, objects, etc. This database aims to close the gap between an object-oriented

and relational database model by using different programming languages, such as C#, C++, and Java.

Transactional Databases

A database management system, also known as a transactional database, can reverse any transacti0n made in the database if it was not performed in the right way. This is a unique capability, which was defined a while back in the form of a relational database. These are the only databases that support such activities.

Pros and Cons of Data Mining

Pros

- Data mining techniques enable organizations to obtain information and trends from the data set. They can use this information to make informed decisions for the organization
- Through data, mining organizations can make the necessary modifications in production and operation
- Data mining is not as expensive as other forms of statistical data analysis
- Businesses can discover hidden trends and patterns in the data set and calculate the probabilities of the occurrence of specific trends
- Since data mining is an easy and quick process, it does not take too long to introduce it onto a new or existing platform. Data mining processes and algorithms can be used to analyze large volumes of data in a short span.

Cons

- Data privacy and security is one of the major concerns of data mining. Organizations can sell their customers' data to other organizations for money. American Express has sold the purchases of their customers to other organizations for money
- Most data mining software uses extensive and difficult algorithms to operate, and any user working on these algorithms needs the required training to work on those models

- Different models work in different ways because of the different algorithms and concepts used in those models. Therefore, it is important to choose the right data mining model
- Some data mining techniques do not produce accurate results, and this can lead to severe repercussions

Applications of Data Mining

Most organizations with intense demands from consumers use data mining. Some of these organizations include communication, retail, marketing company, finance, etc. They use it for the following:

1. To identify customer preferences
2. Understand how customers can be satisfied
3. Assess the impact of various processes on sales
4. The positioning of products in the organization
5. Assess how to improve profits

Retailers can use data mining to develop promotions and products to attract their customers. This section covers some areas where data mining is used widely.

Healthcare

Data mining can improve different aspects of the health system since it uses both analytics and data to obtain better insights from the data sets. The healthcare industry can use this information to identify the right services to improve health care services and reduce costs. Most businesses also use data mining approaches, such as data visualization, machine learning, soft computing, statistics, and multi-dimensional databases, to analyze different data sets and forecast the patients in different categories. These data mining processes enable the healthcare industry to ensure patients obtain the necessary intensive care at the right time and place. Data mining also enables insurers to identify any abuse and fraud.

Market Basket Analysis

This form of analysis is based on different hypotheses. If you purchase specific products, you will probably buy another product from the same

product group. This form of analysis makes it easier for any retailer to identify any customer's purchase behavior in the customer group. The retailer can also use this information to understand what a buyer or customer wants or needs, making it easier for them to alter the store's layout. You can also make a comparison between different stores that make it easier for you to differentiate between different customer groups.

Education

The use of data mining in education is relatively new, and the objective of using data mining in this industry is to explore knowledge from large volumes of data from educational environments. Educational data mining (EDM) can be used to understand the future behavior of a student by studying the impact of various educational systems and support on the student. Educational organizations use data mining to make the right decisions to help students improve. They also use data mining to predict a student’s results. Educational institutions can use this information to identify what a student should be taught. They can also use this information to define how to teach students.

Manufacturing Engineering

Every manufacturing company must know what the customers want, and this knowledge is their asset. You can use various data mining tools to help you identify any hidden patterns and trends in various manufacturing processes. You can also use data mining to develop the right company designs and obtain any correlation between product portfolios and architecture. You can incorporate different customer requirements to develop a model that caters to both the business and customer needs. This information can then be used to forecast product development, cost, and delivery dates, among other criteria.

Customer Relationship Management (CRM)

The objective of CRM is to obtain and maintain customers, thereby enabling businesses to develop customer-oriented strategies and enhance customer loyalty. If you want to improve your relationship with customers, you need to collect the right information and analyze it accurately. When

you use the right data mining technologies, you can use the data collected to analyze and identify methods to improve customer relationships.

Fraud Detection

Have you ever loaned someone money, and they ghosted you immediately after? That is an example of fraud, but this is only on a small scale. Banks and other financial institutions lose close to a billion dollars each year because of fraudulent customers. Traditional fraud detection methods are sophisticated and time-consuming. Data mining techniques and methods use different statistical and mathematical algorithms to identify hidden and meaningful data set patterns. A fraud detection system should be used to protect all the information in the data set while protecting each user's data. Supervised data mining models have a collection of training or sample records using which the model can classify some customers as frauds. You can construct a model using this information. The objective of this model is to identify if there are fraudulent claimants and documents or not.

Lie Detection

It is not difficult to apprehend criminals, but it is extremely difficult to bring the truth out of them. This is a very challenging task. Many police departments and law enforcement agencies now use data mining techniques to minor any communication between suspected terrorists, investigate prior offenses, etc. The data mining algorithms used for this also include text mining. In this process, the algorithm goes through various text files and data to identify hidden patterns in the data set. The data used in this format is often unstructured. These algorithms compare the current output against previous outputs to develop a lie detection model.

Financial Banking

Banks have now taken a turn and have started digitizing all the transactions and information stored by customers. Using data mining algorithms and techniques, bankers can solve various business-related issues and problems. They can use these models to identify various trends, correlations, and patterns in the data collected. Bankers can use these methods when they

work with large volumes of data. It is easier for managers and experts to use these data and correlations to better acquire, target, segment, maintain and retain various customer profiles.

Challenges

Data mining is an important process and extremely powerful. There are, however, many challenges you may face when you implement or execute these algorithms. These challenges are related to data, performance, techniques, and methods used in data mining. The data mining process becomes effective only when you identify the problems or challenges and resolve them effectively.

Noisy and Incomplete Data

As mentioned earlier, the process of extracting useful information and trends from large volumes of data is termed data mining. It is important to remember that data collected in real-time is incomplete, noisy, and heterogeneous. It is difficult to determine if the data collected is reliable or accurate. These are some problems that occur because of human errors or inaccurate measuring instruments. Let us assume you run a retail chain. Your job is to collect the number of every customer who spends more than $1000 at your store. When you identify such a customer, you send a notification to the accounting person, who then enters the information. The accounting person can enter the incorrect number in the data set, which will lead to incorrect data. Some customers may also enter the wrong number in a hurry or for other reasons. Other customers may not want to enter their number for privacy reasons. These situations make the data mining process challenging.

Data Distribution

Real-world data is stored on numerous platforms in a computing environment distributed across different platforms. The data can be stored on the Internet, individual systems, or in a database. This makes it difficult to shift the data from these sources into a central data repository due to different technical and organizational concerns. For instance, some regional offices may have their data stored on their servers to store the data. It is

impossible to store the data from different regional offices on one server if you think about it. Therefore, if you want to mine data, you need to develop the necessary algorithms and tools which make it easier to mine large volumes of data.

Complex Data

Businesses now collect data from different sources, and this data is heterogeneous in nature. It can include different multimedia data, such as video, audio, and images, and other complex data, such as time series, spatial data, and so on. It is difficult for anybody to manage this data and analyze it or extract useful information from it. New tools, methodologies, and technologies must be refined most times if you want to obtain the required information.

Performance

The performance of any data visualization model relies on the algorithm being used and its efficiency. The technique with which the model is developed also determines the performance of the model. If the algorithm designed is not built correctly, the efficiency of the process is significantly affected.

Data Security and Privacy

Data mining can lead to a serious issue when it comes to data governance, privacy and security. Let us assume you are a retailer who analyzes a customer’s purchasing patterns. To do this, you need to collect all your customers' data, purchasing habits, preferences, and other details. You need to collect this information, and you may not require your customer’s permission to do this.

Data Visualization

Data visualization is an important process in data mining. This is the only way you or a business can visualize the different patterns and trends in the data set. Businesses and data scientists need to identify what the data and variables in the data set are trying to convey. It is also important to know

what the data is trying to express to you. There are times when it is not easy to present the data in an easy-to-understand manner. Some input data points, or variables, may produce complicated outputs. Therefore, you need to identify efficient and accurate data visualization processes if you want to succeed.

Chapter Two: Data Mining Architecture

Now that you have a basic idea of data mining, let us learn more about the data mining architecture. Some significant components of data mining are the data mining engine, data source, the pattern evaluation module, knowledge base, data warehouse servers, and graphical user interface. Let us look at each of these components in further detail.

Data Source

Data can be sourced from the following:

- Database
- Data warehouse
- Internet
- Text files

If you want to obtain useful information from data mining, you need to collect large volumes of historical information. Most organizations store data in data warehouses or databases. A data warehouse can include more than one database, a data repository, or text file spreadsheets. You can also use spreadsheets or text files as the source since they can contain some information.

Different Processes

Before the collected data or information is moved into a data warehouse or database, the information should be processed, selected, integrated, and cleaned. Since information is collected from numerous sources that store data in different formats, you cannot use it directly to perform any data mining operations. The results of the data mining process will be inaccurate and incomplete if you use unstructured data. Therefore, the first step in the process is to clean the data that you need to work with and then pass it onto the server. The process of cleaning the data is not as easy as one thinks. You can perform different kinds of operations on the data as part of the cleaning, integration, and selection.

Data Warehouse Server or Database

Once you select the data you want to use from different sources, you can clean it and pass it onto the data warehouse server or database. This is the source of the original data that you will process and use in the data mining process. The user uses the server, meaning you or the business, to retrieve the information relevant to the data mining request.

Data Mining Engine

This is a very important component of the data mining architecture since it contains different modules that can be used to perform various data mining tasks. These include:

- Classification
- Characterization
- Association
- Prediction
- Clustering
- Time-series analysis

In simple words, we can say that this engine is the root or base of the entire architecture. The engine comprises different software and instruments that can be used to obtain knowledge and insights from the mining process's data. You can also use the engine to learn more about any kind of data stored in the server.

Pattern Evaluation Module

This model is used in the data mining architecture to measure or investigate the pattern, followed by the variables based on a threshold value. This module works with the data mining engine to identify various patterns in the data set. The pattern evaluation module uses different stake measures that cooperate with various data mining modules in the engine to identify different patterns or trends in the data sets. This module uses a stake threshold to locate any hidden patterns and trends in the data set.

The pattern evaluation module can work with the mining module, but this is only dependent on the different techniques used in the data mining engine. If you want to develop efficient and accurate data mining models, you need to push the evaluation of this stake measure as much as possible into the

mining process. This will ensure the model only looks at the different patterns in the data set.

Graphical User Interface (GUI)

The GUI is one of the data mining architecture modules that communicate between the user and the data mining system or module. This module helps users to efficiently and easily communicate with the system without worrying about how complex the process is. The GUI module works with the data mining system based on the user's task or query to display the required results.

Knowledge Base

This is the last module in the data set, which helps the entire data mining process. This module can be used to evaluate the stake measure used to identify hidden results and guide the search. The knowledge base module contains data from a user’s experience, user views, and other information, which helps in the data mining process. The knowledge base obtains inputs and information from the data mining engine to obtain reliable and accurate information. The knowledge base also interacts with the pattern assessment module to obtain inputs and also update the data stored in the module.

Chapter Three: Data Mining Techniques

Now, let us look at some data mining techniques, which can be incorporated into the data mining engine. These techniques allow you to identify hidden, valid, and unknown patterns and correlations in the large data sets. These techniques use different machine learning techniques, mathematical algorithms, and statistical models to answer different questions. Some examples of such algorithms are neural networks, decision trees, classification, etc. Data mining predominantly uses prediction and analysis.

Data mining professionals use different methods to understand, process, and analyze data to obtain accurate conclusions from large volumes of data. The methods they use are dependent on various technologies and methods from the intersection of statistics, machine learning, and database management. So, what are the methods they use to obtain these results?

In most data mining projects, professionals have used different data mining techniques. They have also developed and used different modules and techniques, such as classification, association, prediction, clustering, regression, and sequential patterns. We will look at some of these in further detail in this chapter.

Classification

The classification technique is used to obtain relevant and important information about the metadata and data used in the mining process. Professionals use this technique to classify data points and variables into different classes. Some of these techniques can be classified into the following:

1. We can classify various data mining frameworks and techniques based on the source of data you are trying to mine. This process is based on the data you use or handle. For instance, you can classify data into time-series data, text data, multimedia, World Wide Web, spatial data, etc.
2. Data can be classified into different frameworks based on the database you use in your analysis. This type of classification is based on the type of model you are using. For instance, you can

classify the data into the following categories: relational database, object-oriented database, transactional database, etc.

3. We can classify data into a framework based on the type of knowledge extracted from the data set. This form of classification is dependent on the type of information you have extracted from the data. You can also use the different functionalities used to perform this classification. Some frameworks used are clustering, classification, discrimination, characterization, etc.
4. Data can also be classified into a framework based on the different techniques used to perform data mining. This form of classification is based on the approach of analysis used to mine the data, such as machine learning, neural networks, visualization, database-oriented and data warehouse-oriented, genetic algorithms, etc. This form of classification also takes into account how interactive the GUI is.

Clustering

Clustering is an algorithm used to divide information into groups of connected objects based on their characteristics. When you divide the data set into clusters, you may lose some details present in the data set, but you improve the data set. When it comes to data modeling, clustering is rooted in mathematics, numerical analysis, and statistics. If you look at data modeling in terms of machine learning, the clusters show hidden patterns in the data set. The model looks for clusters in the data set using unsupervised machine learning. The subsequent framework developed will represent a concept of data. When you look at it from a practical point of view, this form of analysis plays an important role in various data mining applications, such as text mining, scientific data exploration, spatial database applications, Web analysis, CRM, computational biology, information retrieval, medical diagnostics, etc.

In simple words, clustering analysis is a form of data mining technique used to identify the data points in the data set, which share numerous similarities. This technique will help you recognize various similarities and differences in the data set. Clustering is similar to classification, but in this technique, you group large chunks of data into groups based on the similarities.

Regression

Regression analysis is another form of data mining, which is used to analyze and identify the relationship between different variables based on the presence of another variable in the data set. This technique is used to define the probability of the occurrence of a specific variable in the data set. The process of regression is a form of modeling and planning used in different algorithms. For instance, you can use this technique to project a cost or expense based on various factors, such as consumer demand, competition, and availability. This technique will give you the exact relationship between the variables in the data set. Some forms of this technique are linear regression, multiple regression, logistic regression, etc.

Association Rules

Data mining's association technique is to define a link between various data points in the data set. Using this technique, data mining professionals can identify any hidden patterns or trends in the data set. An association rule is a conditional statement using the if-then format, and these rules help the professional identify the probability of interactions between different data points in large data sets. You can also identify correlations between different databases as well.

The association rule mining technique is used in different applications and is often used by retailers to identify correlations in medical data sets or retail data sets. The algorithm works differently on different data sets. For instance, you can collect the data of all the items you purchased in the last few months and run some association rules on the items to see what you want to purchase together. Some measurements used are:

Lift

This measurement is used to define the accuracy of the probability of how often you have purchased a specific product. The formula used to do this: (confidence interval) / (item A) / (entire data set).

Support

This process is used to determine how often you purchase different items and compares that to the overall data collected. The formula used to do this

is: (item C + item D) / (entire data set).

Confidence

This process measures the number of times you purchase a specific product when you purchase another product as well. To do this, you can use the following formula: (item C + item D) / (item D)

Outer Detection

The outer detection technique is used when you need to identify the patterns or data points in the data set that do not match the expected data set behavior or pattern. This technique is often used in domains such as detection, intrusion, fraud detection, etc. Outer detection is also known as outlier mining or outlier analysis.

An outlier is any point in the data set that does not behave in the same way as the average set of data points in the dataset. Most data sets have outliers in them, and this should be expected. This technique plays an important role in the field of data mining. This technique is used in different fields, such as debit or credit card fraud detection, detection of outliers in wireless sensor network data, network interruption identification, etc.

Sequential Patterns

A sequential pattern is another technique used in data mining, which specializes in the evaluation of sequential information in the data set. This is one of the best ways to identify any hidden sequential patterns in the data set. This technique uses different subsequences from a single sequence where the stake cannot be measured in terms of various criteria, such as occurrence frequency, length, etc. In simple words, this process of data mining allows users to recognize or discover different patterns, either small or big, in the transaction data over a certain period.

Prediction

This technique uses a combination of different data mining processes and techniques, such as clustering, trends, classification, etc. Prediction looks at

historic data, in the form of instances and events, in the appropriate sequence to predict the future.

Chapter Four: Data Mining Tools

As mentioned earlier, data mining uses a set of techniques with specific algorithms, statistical analysis, database systems, and artificial intelligence to analyze and assess data from various data sources and from different perspectives. Most data mining tools can discover different groupings, trends, and patterns in a large data set. These tools also transform the data into refined information.

Some of these frameworks and techniques allow you to perform different actions and activities that are key for data mining analysis. You can perform different algorithms, such as classification and clustering, on your data set using these data mining tools and techniques. The techniques use a framework, which provides insights for the data and various phenomena represented by the data set. These frameworks are termed data mining tools.

This chapter covers some common data mining tools used in the industry.

Orange Data Mining

Orange is a suite which uses different machine learning and data mining software. It also supports data visualization and is a software component, which is written in Python. This application was developed by the faculty of information and computer science from Ljubljana University, Slovenia. Since this is an application that uses software-based components, the application's components are termed widgets. The different widgets used in the application can be used for preprocessing and data visualization. Using these widgets, you can assess different algorithms for data mining and also use predictive modeling. These widgets have different functionalities, such as

- Data reading
- Displaying data in the form of tables
- Selection of certain features from the data set
- Comparison between different learning algorithms
- Training predictors
- Data visualization

Orange also provides an interactive interface which makes it easier for users to work with different analytical tools. It is easy to operate the applications.

Why Should You Use Orange?

If you have data you collect from different sources, it can be formatted and arranged quickly in this application. You can format the data, so it follows the required pattern and move the widgets around to improve the interface. Different users can use this application since it allows a user to make a smart decision in a short time. You can do this by analyzing and comparing data. Orange is a great way to visualize open-source data. It also enables users to evaluate different data sets. You can perform data mining using different programming languages, including Python and visual programming. You can perform different analyses on this platform.

The application also comes with machine learning components, text mining features, and add-ons for bioinformatics. It also comes with different data analytics features and comes with a python library.

You can run different Python scripts on a terminal window or use an integrated environment, such as PythonWin, PyCharmand, and pr shells like iPython. The application has a canvas interface on which you can place the widget. You can then create a workflow in the interface to analyze the data set. The widget can also perform fundamental operations, such as showing a data table, reading the data, training predictors, selecting required features from the data set, visualizing data elements, comparing learning algorithms, etc. The application works on various Linux operating systems, Windows and Mac OS X. It also comes with classification and multiple regression algorithms.

This application also reads documents in their native or other formats. It uses different machine learning techniques to classify the data into categories or clusters to aid in supervised data mining. Classification algorithms use two types of objects, namely classifiers and learners. A learner is termed class-leveled data, and it uses the data to return the classifier. You can also use regression methods in orange, and these are similar to classification methods; both techniques are designed for supervised data mining, and they need class-level data. The ensemble will continue to learn using a combination of predictions from the individual

model and precision measures. The model you develop can come from using different learners or different training data on the same data sets.

You can also diversify learners by changing the parameter set used by the learners. In this application, you can create ensembles by wrapping them around different learners. These ensembles also act like other learners, but the results they return allow you to predict the results for any other data instance.

SAS Data Mining

The SAS institute developed SAS or the Statistical Analysis System, and it is used for data management and analytics. You can use SAS to mine data, manage information from different sources, change the data and analyze the data's statistics. If you are a non-technical user, you can also use the graphical user interface to communicate with the application. The SAS data miner analyzes large volumes of data that provide the user with accurate insights to make the right decisions. SAS uses distributed memory processing architecture, which can be scaled in different ways. This application can be used for data optimization, mining, and text mining.

DataMelt Data Mining

DataMelt, also known as DMelt, is a visualization and computation environment that offers users an interactive structure that can be used for data visualization and analysis. This application was designed especially for data scientists, engineers, and students. This application uses a multi-platform utility and is written in Java programming language. You can run this application on different operating systems as long as they are compatible with a Java Virtual Machine (JVM). This application consists of mathematics and science libraries.

- **Mathematical Libraries** : The libraries used in this application are used for algorithms, random number generation, curve fitting, etc.
- **Scientific Libraries** : The libraries used are for drawing 3D or 2D plots.

DMelt can be used to analyze large volumes of data, statistical analysis, and data mining. This application is used in financial markets, engineering, and natural sciences.

Rattle

This tool or application is a tool that uses a graphic user interface (GUI). Rattle is developed using the R programming language. The application also exposes R's statistical power and offers different data mining features that can be used during the mining process. Rattle has a well-developed and comprehensive user interface and includes an integrated log tab, allowing users to produce the code to perform different GUI operations. You can use the application to produce data sets, and you can edit and view them. The application also allows you to review the code, use it for different purposes, and extend that code without any restrictions.

Rapid Miner

Most data mining professionals use rapid miner to perform predictive analytics. This tool was developed by a company named Rapid Miner. The code is written in Javascript language, and it offers users an integrated environment they can use to perform various operations apart from predictive analysis, such as deep learning, machine learning, and text mining. Rapid miner can be used in different applications, such as commercial applications, education, research, training, machine learning, application development, and company applications. Rapid miner also provides users with a server on-site. It also allows users to use both private or public cloud infrastructure to store the data and perform operations on that data set. The base of this application is a client/server model. This application is relatively accurate when compared to other applications and tools and uses a template-based framework.

Chapter Five: Introduction to Data Analysis

Now that you have an idea of what data mining is, let us understand what data analysis is and the different processes used in data analysis in brief. We will look at these concepts in further detail later in the book.

Data analysis is the process of transforming, cleaning, and modeling any information collected to identify hidden patterns and information in the data set to make informed decisions. Data analysis aims to extract useful information hidden in the data set and take the required decision based on the results of the analysis.

Why Use Data Analysis?

If you want to grow in life or improve your business, you need to perform some analysis on the data collected. If your business does not grow, you need to go back and acknowledge what mistakes you made and overcome those mistakes. You also need to find a way to prevent these mistakes from happening again. If your business is growing, you need to look forward to making the necessary changes to your processes to make sure the businesses grow more. You need to analyze the information based on the business processes and data.

Data Analysis Tools

You can use different data analysis tools to manipulate and process data. These tools make it easier for you to analyze the correlations and relationships between different data sets. These tools also make it easier for you to identify any hidden insights or trends in the data set.

Data Analysis Types

The following are the different forms of data analysis.

Text Analysis

This form of analysis is also called data mining, and we have looked at this in great detail in the previous chapters.

Statistical Analysis

Using statistical analysis, you can define what happened in a certain event based on historical information. This form of analysis includes the following processes:

1. Collection
2. Processing of information
3. Analysis
4. Interpretation of the results
5. Presentation of the results
6. Data modeling

Statistical analysis allows you to analyze sample data. There are two forms of statistical analysis:

Descriptive Analysis

In this form of analysis, you look at the entire data set or only a portion of the summarized data set in the form of numerals. You can use this numerical information to calculate the values of central tendency.

Inferential Analysis

In this analysis, you look at a sample from the entire data set. You can select different samples and perform the same processes to determine how the data set is set. This form of analysis also tells you how the data set is structured.

Diagnostic Analysis

Diagnostic analysis is used to determine how a certain event occurred. This type of analysis uses statistical models to identify any hidden patterns and insights from the data set. You can use diagnostic analysis to identify any new business process problems and see what caused that problem. You can also identify any similar patterns within the data set to see if there were any other problems with similar patterns. This form of analysis enables you to use prescriptions for any new problems.

Predictive Analysis

Predictive analysis is the use of historic data to determine what may happen in the future. The simplest example is where you decide what purchases you want to make. Let us assume you love shopping and bought four dresses after dipping into your savings. Now, if your salary were to double the next year, you can probably buy eight dresses. This is an easy example, and not every analysis you do will be this easy. You need to think about the various circumstances when you perform this analysis since the prices of clothes can change over the next few months.

Predictive analysis can be used to predictions about future outcomes based on historical and current data. It is important to note that the results obtained are only forecasts. The accuracy of the model used is dependent on the information you have and how you can dig into it.

Prescriptive Analysis

The process of prescriptive analysis uses a combination of the insights and results of previous analyses and the action you want to take to solve a current decision or problem. Most companies are now data-driven, and they use this form of analysis since they need both descriptive and prescriptive analyses to improve data performance. Data professionals use these technologies and tools to analyze the data they have and derive the results.

Data Analysis Process

The process followed for data analysis is solely dependent on the information you gather and the application or tool you use to analyze and explore the data. You need to find patterns in the data set. Based on the data and information you collect, you can make the necessary information to obtain the ultimate result or conclusion. The process followed is:

- Data Requirement Gathering
- Data Collection
- Data Cleaning
- Data Analysis
- Data Interpretation
- Data Visualization

Data Requirement Gathering

When it comes to data analysis, you need to determine why you want to perform this analysis. The objective of this step is to determine what the aim of your analysis is. You should decide what type of analysis you want to perform and how you want to perform this analysis. In this step, you should determine what you want to analyze and how you plan to measure or analyze this information. It is important to determine why you need to investigate and identify the measures you want to use to perform this analysis.

Data Collection

After you gather the requirement, you will obtain a clear idea of what data you have and what you need to measure. You will also know what to expect from your findings. It is now time for you to collect your data based on the requirements of your business. When you collect the data from the sources, you need to process and organize it before you analyze it. Since you collect data from different sources, you need to maintain a log with the date of collection and information about the source.

Data Cleaning

The data you collect may be irrelevant for you or may not be useful for your analysis. Therefore, you need to clean it before you perform these processes. The data may contain white spaces, errors, and duplicate records, and thus it should be cleaned and free of errors. This should be done before you analyze the data because your analysis results are based on how well you clean the data.

Data Analysis

When you collect, process, and clean data, you can analyze it. When you manipulate data, you need to find a way to extract the information from the data set. If you do not find the necessary information, you need to collect more information from the data set. During this phase of the process, you can use the tools, techniques, and software for data analysis, which will

enable you to interpret, understand, analyze, and extract necessary conclusions based on the requirement.

Data Interpretation

When you analyze the data completely, it is time for you to interpret the results. When you have the results, you can either use a chart or table to display the analysis. You can use the results of the analysis to identify the best action you can take.

Data Visualization

Most people use data visualization regularly, and they use often appear in the form of graphs and charts. In simple words, when you show data in the form of a graph, it is easier for the brain to understand the information and process it. Data visualization is used to identify any hidden facts, trends, and correlations in the data set. When you observe the relationships or correlations between the data points, you can obtain meaningful and valuable information.

Chapter Six: Manipulation of Data in Python

Data processing and cleaning is an important aspect of data analysis. This chapter sheds some light on the different ways you can use the Pandas and NumPy libraries to manipulate the data set.

NumPy

#Using the sections below, we can check the library version to determine we are not using an old version

import numpy as np

np.__version__

'1.12.1'

#The following statements create a list with numbers from 0 to 9

L = list(range(10))

#The next few lines of code convert integers to strings. This process is called list comprehension. This process is one of the best ways to handle list manipulations.

[str(c) for c in L]

['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']

[type(item) for item in L]

[int, int, int, int, int, int, int, int, int, int]

Create Arrays

Arrays are homogeneous types of data. If you are familiar with programming languages, you will have an idea of how you can use arrays. Arrays only hold specific variables in the data set, and this is true in any programming language.

#creating arrays

np.zeros(10, dtype='int')

array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0])

#We will now create a 3 x 5 array

np.ones((3,5), dtype=float)

array([[1., 1., 1., 1., 1.],

1., 1., 1., 1., 1.],

1., 1., 1., 1., 1.]])

#The next lines of code are used to create arrays with a predefined value np.full((3,5),1.23)

array([[1.23, 1.23, 1.23, 1.23, 1.23],

1.23, 1.23, 1.23, 1.23, 1.23],

1.23, 1.23, 1.23, 1.23, 1.23]]) #We are creating an array using the sequence np.arange(0, 20, 2)

array([0, 2, 4, 6, 8,10,12,14,16,18])

#The next few lines of code are used to create arrays with variables with the space between the values np.linspace(0, 1, 5)

array([0., 0.25, 0.5 , 0.75, 1.])

#We will now create a 3 x 3 array with the values of mean and standard deviation as 0 and 1, respectively. We will use the dimension np.random.normal(0, 1, (3,3))

array([[0.72432142, -0.90024075, 0.27363808],

0.88426129, 1.45096856, -1.03547109], [-0.42930994, -1.02284441, -1.59753603]]) #create an identity matrix

np.eye(3) array([[1., 0., 0.],

0., 1., 0.],

0., 0., 1.]])

#set a random seed np.random.seed(0)

x1 = np.random.randint(10, size=6) #one dimension

x2 = np.random.randint(10, size=(3,4)) #two dimension

x3 = np.random.randint(10, size=(3,4,5)) #three dimension

```
print("x3 ndim:", x3.ndim)
print("x3 shape:", x3.shape)
print("x3 size: ", x3.size)
('x3 ndim:', 3)
('x3 shape:', (3, 4, 5))
('x3 size: ', 60)
```

Array Indexing

It is important to remember the index of the array only begins at zero. The first value in an array has the index zero.

```
x1 = np.array([4, 3, 4, 4, 8, 4])
x1
array([4, 3, 4, 4, 8, 4])
#assess value to index zero
x1[0]
4
#assess fifth value
x1[4]
8
#get the last value
x1[-1]
4
#get the second last value
x1[-2]
8
#We are specifying the rows and columns for a multidimensional array
array([[3, 7, 5, 5],
```

```
[0, 1, 5, 9],
[3, 0, 5, 0]])
#1st row and 2nd column value
x2[2,3]
0
#3rd row and last value from the 3rd column
x2[2,-1]
0
#replace value at 0,0 index
x2[0,0] = 12
x2
array([[12, 7, 5, 5],
0, 1, 5, 9],
3, 0, 5, 0]])
```

Array Slicing

Most programming languages allow you to access a range of elements in the array. To do this, you would need to slice the array into parts.

```
x = np.arange(10)
x
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
#from the first to the fourth position
x[:5]
array([0, 1, 2, 3, 4])
#from the fourth to the last element
x[4:]
array([4, 5, 6, 7, 8, 9])
```

#from the fourth to the sixth position

x[4:7]

array([4, 5, 6])

#The code returns the elements at even indices

x[: : 2]

array([0, 2, 4, 6, 8])

#These lines return elements in intervals of two from the first element

x[1::2]

array([1, 3, 5, 7, 9])

#reverse the array

x[::-1]

array([9, 8, 7, 6, 5, 4, 3, 2, 1, 0])

Array Concatenation

Most programmers split the data set into multiple arrays, but they would need to combine or concatenate these arrays to perform different operations. You do not have to type the elements in the different arrays but can combine them to handle the operations easily.

#Using the lines of code below, you can concatenate more than two arrays at once

x = np.array([1, 2, 3])

y = np.array([3, 2, 1])

z = [21,21,21]

np.concatenate([x, y, z])

array([1, 2, 3, 3, 2, 1, 21, 21, 21])

#The following functions can be used to create two-dimensional arrays

grid = np.array([[1,2,3],[4,5,6]])

np.concatenate([grid,grid])

array([[1, 2, 3],

[4, 5, 6],

[1, 2, 3],

[4, 5, 6]])

#You can define the column and row wise matrix using the axis parameter np.concatenate([grid,grid],axis=1)

array([[1, 2, 3, 1, 2, 3],

[4, 5, 6, 4, 5, 6]])

In the code written above, the concatenation function has been used on the arrays. It is important to note that the arrays with the same data types are combined for these operations.

You can create different types of arrays based on the number of dimensions. But how do you combine arrays of different dimensions together? The next few lines of code use the np.vstack and the np.hstack (similar to the VLookUp and HLookUp functions in excel) to combine the data.

x = np.array([3,4,5])

grid = np.array([[1,2,3],[17,18,19]])

np.vstack([x,grid])

array([[3, 4, 5],

1, 2, 3], [17, 18, 19]])

#The following keywords and functions can be used to add arrays np.hstack z = np.array([[9],[9]])

np.hstack([grid,z]) array([[1, 2, 3, 9], [17, 18, 19, 9]])

You can split the array using pre-defined positions or criteria.

x = np.arange(10)

x

array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]) x1,x2,x3 = np.split(x,[3,6]) print x1,x2,x3

[0 1 2] [3 4 5] [6 7 8 9]

grid = np.arange(16).reshape((4,4)) grid

upper,lower = np.vsplit(grid,[2]) print (upper, lower)

(array([[0, 1, 2, 3],

[4, 5, 6, 7]]), array([[8, 9, 10, 11], [12, 13, 14, 15]]))

The NumPy directory in Python gives you access to different mathematical functions, which can be used in Python. Some of these functions include, sum, divide, abs, multiple, mod, power, log, sin, tan, cos, mean, min, max, var, etc. You can use one or more of these functions to perform arithmetic calculations. If you want to learn more about the use of these functions, you can read the NumPy documentation to learn more about the various functions.

Now that we have looked at how we can use the NumPy library let us look at the Pandas library and see how it can be used. Make sure to read each line of code before you manipulate the information you have collected.

Let's Start with Pandas

import pandas as pd

#We will create a data frame dictionary to cover the names of rows and columns

data = pd.DataFrame({'Country': ['Russia','Colombia','Chile','Ecuador','Nigeria'],

Rank':[121,40,100,130,11]})

data

CountryRank

0Russia121

1Colombia40

2Chile100

3Equador130

4Nigeria11

#The following lines of code produce an analysis of the data

data.describe()

Rank

count5.000000

mean80.400000

std52.300096

min11.000000

25%40.000000

50%100.000000

75%121.000000

max130.000000

How to Summarize Data Using Pandas

The Pandas library has different functions that enable you to obtain a statistical summary of all the data set variables. You can do this using the describe function or method in the library. If you want to look at all the information present in the data set, you can use the function or method info().

#Among other things, it shows the data set has five rows and two columns with their respective names. data.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 5 entries, 0 to 4

Data columns (total two columns):

Country 5 non-null object

Rank 5 non-null int64

dtypes: int64(1), object(1)

memory usage: 152.0+ bytes

#We will now create another data frame

data = pd.DataFrame({'group':['a', 'a', 'a', 'b', 'b', 'b', 'c', 'c', 'c'],'ounces':[4, 3, 12, 6, 7.5, 8, 3, 5, 6]})

data

groupounces

0a4.0

1a3.0

2a12.0

3b6.0

4b7.5

5b8.0

6c3.0

7c5.0

8c6.0

#We will sort the data frame by making changes to the data set using the following functions: inplace = True and data.sort_values(by= ['ounces'],ascending=True,inplace=False) groupounces

1a3.0

6c3.0

0a4.0

7c5.0

3b6.0

8c6.0

4b7.5

5b8.0

2a12.0

You can now sort the data using different columns.

data.sort_values(by=['group','ounces'],ascending=
[True,False],inplace=False)

groupounces

2a12.0

0a4.0

1a3.0

5b8.0

4b7.5

3b6.0

8c6.0

7c5.0

6c3.0

How to Work with Noise or Duplicate Records

Every data set has duplicate columns and rows that are often termed noise. As mentioned earlier, you need to get rid of this noise if you want to obtain accurate results from the analysis. Never feed a model with unprocessed data. This section sheds some light on how you can remove any duplicates in the data set.

#create another data with duplicated rows

data = pd.DataFrame({'k1':['one']*3 + ['two']*4, 'k2':[3,2,1,3,3,4,4]})

data

k1k2

0one3

1one2

2one1

3two3

4two3

5two4

6two4

#sort values

data.sort_values(by='k2')

k1k2

2one1

1one2

0one3

3two3

4two3

5two4

6two4

#remove duplicates - ta-da!

data.drop_duplicates()

k1k2

0one3

1one2

2one1

3two3

5two4

In the above example, we are mapping the values present in the columns and rows to remove any duplicate values present in the data set. You can remove these duplicates using certain columns as parameters. The following lines of code look at how we can remove duplicates from specific columns.

data.drop_duplicates(subset='k1')

k1k2

0one3

3two3

Categorization of Data

Now that you have removed the duplicates and noise in the data set let us look at how you can categorize the data based on some predefined criteria or rules. You need to do this often if you want to process the data into categories before you run the data through the model. Let us assume you work on data that has information for each country. You may want to split the data based on the continents. If you want to do this, you should create a new variable. The code below can be used to achieve this.

data = pd.DataFrame({'food': ['bacon', 'pulled pork', 'bacon', 'Pastrami','corned beef', 'Bacon', 'pastrami', 'honey ham', 'nova lox'],

'ounces': [4, 3, 12, 6, 7.5, 8, 3, 5, 6]})

data

foodounces

0bacon4.0

1pulled pork3.0

2bacon12.0

3Pastrami6.0

4corned beef7.5

5Bacon8.0

6pastrami3.0

7honey ham5.0

8nova lox6.0

Let us now try to map one animal to another based on each animal's source of food. To do this, we will create a new variable. This variable will store the mapping. If you want the model to perform the mapping, you need to create a dictionary to map the food to the animals. Use a map function to

pull every variable from the dictionary. The following lines of code will help you do this.

```
meat_to_animal = {
'bacon': 'pig',
'pulled pork': 'pig',
'pastrami': 'cow',
'corned beef': 'cow',
'honey ham': 'pig',
'nova lox': 'salmon'
}
def meat_2_animal(series):
if series['food'] == 'bacon':
return 'pig'
elif series['food'] == 'pulled pork':
return 'pig'
elif series['food'] == 'pastrami':
return 'cow'
elif series['food'] == 'corned beef':
return 'cow'
elif series['food'] == 'honey ham':
return 'pig'
else:
return 'salmon'
#create a new variable
data['animal'] = data['food'].map(str.lower).map(meat_to_animal)
data
```

foodouncesanimal

0bacon4.0pig

1pulled pork3.0pig

2bacon12.0pig

3Pastrami6.0cow

4corned beef7.5cow

5Bacon8.0pig

6pastrami3.0cow

7honey ham5.0pig

8nova lox6.0salmon

#Alternatively, you can apply the function once you convert the food values into lower case

lower = lambda x: x.lower()

data['food'] = data['food'].apply(lower)

data['animal2'] = data.apply(meat_2_animal, axis='columns')

data

foodouncesanimalanimal2

0bacon4.0pigpig

1pulled pork3.0pigpig

2bacon12.0pigpig

3pastrami6.0cowcow

4corned beef7.5cowcow

5bacon8.0pigpig

6pastrami3.0cowcow

7honey ham5.0pigpig

8nova lox6.0salmonsalmon

You can also use the assign function to create new variables. This function uses different Pandas functions that you can use while you work with different data sets. You need to keep these functions in mind.

How to Solve Problems Using Pandas

The next section sheds some light on how you can use the Pandas library to solve different problems. The Pandas library has different packages that allow you to handle big data sets.

data.assign(new_variable = data['ounces']*10)

foodouncesanimalanimal2new_variable

0bacon4.0pigpig40.0

1pulled pork3.0pigpig30.0

2bacon12.0pigpig120.0

3pastrami6.0cowcow60.0

4corned beef7.5cowcow75.0

5bacon8.0pigpig80.0

6pastrami3.0cowcow30.0

7honey ham5.0pigpig50.0

8nova lox6.0salmonsalmon60.0

Let us now remove the animal2 column from the data frame.

data.drop('animal2',axis='columns',inplace=True)

data

foodouncesanimal

0bacon4.0pig

1pulled pork3.0pig

2bacon12.0pig

3Pastrami6.0cow

4corned beef7.5cow

5Bacon8.0pig

6pastrami3.0cow

7honey ham5.0pig

8nova lox6.0salmon

How to Clean Data Sets

Most organizations use different sources to collect information. These data sets will need to be cleaned and processed. You may find some missing values in the data set. If you want to remove any missing data from the data set, you need to substitute the missing value with a dummy variable. You can also use a default value if you would like to. There is a possibility there are outliers in the data set, and you need to remove them. Let us see how this can be done in the directory.

#In the sections below, we will use a series function from the library to create an array

data = pd.Series([1., -999., 2., -999., -1000., 3.])

data

0 1.0

-999.0

2 2.0

3 -999.0

4 -1000.0

5 3.0

dtype: float64

#replace -999 with NaN values data.replace(-999, np.nan,inplace=True) data

0 1.0

NaN

2.0

3 NaN

4 -1000.0

5 3.0

dtype: float64

#The following lines of code allow you to replace multiple values at once.
data = pd.Series([1., -999., 2., -999., -1000., 3.])
data.replace([-999,-1000],np.nan,inplace=True) data

0 1.0

1 NaN

2.0

NaN

4 NaN

5 3.0

dtype: float64

The next few lines of code will help you rename columns and rows.

data = pd.DataFrame(np.arange(12).reshape((3, 4)),index=['Ohio', 'Colorado', 'New York'],columns=['one', 'two', 'three', 'four'])

data

onetwothreefour

Ohio0123

Colorado4567

New York891011

Using rename function

data.rename(index = {'Ohio':'SanF'}, columns=
{'one':'one_p','two':'two_p'},inplace=True) data

one_ptwo_pthreefour

SanF0123

Colorado4567

New York891011

#We can perform the same functions using string

data.rename(index = str.upper, columns=str.title,inplace=True)

data

One_pTwo_pThreeFour

SANF0123

COLORADO4567

NEW YORK891011

Now, we need to categorize or split the continuous variables.

ages = [20, 22, 25, 27, 21, 23, 37, 31, 61, 45, 41, 32]

we will now divide the large data set into smaller bins or segments.

bins = [18, 25, 35, 60, 100]

cats = pd.cut(ages, bins)

cats

[(18, 25], (18, 25], (18, 25], (25, 35], (18, 25], ..., (25, 35], (60, 100], (35, 60], (35, 60], (25, 35]] Length: 12

Categories (4, object): [(18, 25] < (25, 35] < (35, 60] < (60, 100]] #This statement includes the bin values to the right pd.cut(ages,bins,right=False)

[[18, 25), [18, 25), [25, 35), [25, 35), [18, 25), ..., [25, 35), [60, 100), [35, 60), [35, 60), [25, 35)] Length: 12

Categories (4, object): [[18, 25) < [25, 35) < [35, 60) < [60, 100)] #The pandas library assigns each categorical variable with an encoder. cats.labels

array([0, 0, 0, 1, 0, 0, 2, 1, 3, 2, 2, 1], dtype=int8)

#The following will show the number of observations falling under a certain range

pd.value_counts(cats)

(18, 25] 5

(35, 60] 3

(25, 35] 3

(60, 100] 1

dtype: int64

We can also pass a name to every label.

bin_names = ['Youth', 'YoungAdult', 'MiddleAge', 'Senior'] new_cats = pd.cut(ages, bins,labels=bin_names)

pd.value_counts(new_cats)

Youth5

MiddleAge3

YoungAdult3

Senior1

dtype: int64

#using the following lines of code, you can calculate the cumulative sum

pd.value_counts(new_cats).cumsum()

Youth5

MiddleAge3

YoungAdult3

Senior1

dtype: int64

The pandas directory also has libraries which allow you to create pivots or group variables into clusters. It offers easy-to-use pivot tables to perform these forms of clustering. It is for this reason you need to understand and learn how you can do this.

df = pd.DataFrame({'key1' : ['a', 'a', 'b', 'b', 'a'],

'key2' : ['one', 'two', 'one', 'two', 'one'],

'data1' : np.random.randn(5),

'data2' : np.random.randn(5)})

df

data1data2key1key2

00.9735990.001761a

10.207283-0.990160a

21.0996421.872394b

30.939897-0.241074b

40.6063890.053345a

#calculate the mean of data1 column by key1

grouped = df['data1'].groupby(df['key1'])

grouped.mean()

key1

0.595757 b 1.019769

Name: data1, dtype: float64

#We will now slice the data frame

dates = pd.date_range('20130101',periods=6)

df = pd.DataFrame(np.random.randn(6,4),index=dates,columns=list('ABCD')) df

ABCD

2013-01-011.030816-1.2769890.837720-1.490111 2013-01-02-1.070215-0.2091290.604572-1.743058

2013-01-031.5242271.8635751.2913781.300696

2013-01-040.918203-0.158800-0.964063-1.990779

2013-01-050.0897310.114854-0.5858150.298772

2013-01-060.2222600.435183-0.0457480.049898 #this line gets the first n rows from the data set df[:3]

ABCD

2013-01-011.030816-1.2769890.837720-1.490111

2013-01-02-1.070215-0.2091290.604572-1.743058

2013-01-031.5242271.8635751.2913781.300696

#the array is sliced based on the dates

df['20130101':'20130104']

ABCD

2013-01-011.030816-1.2769890.837720-1.490111

2013-01-02-1.070215-0.2091290.604572-1.743058

2013-01-031.5242271.8635751.2913781.300696

2013-01-040.918203-0.158800-0.964063-1.990779 #This line is used to slice the code based on column names df.loc[:,['A','B']]

AB

2013-01-011.030816-1.276989

2013-01-02-1.070215-0.209129

2013-01-031.5242271.863575

2013-01-040.918203-0.158800

2013-01-050.0897310.114854

2013-01-060.2222600.435183

#we now slice the array based on row and column labels
df.loc['20130102':'20130103',['A','B']] AB

2013-01-02-1.070215-0.209129

2013-01-031.5242271.863575

#we are slicing the array based on column indices

df.iloc[3] #returns 4th row (index is 3rd)

0.918203 B -0.158800 C -0.964063 D -1.990779

Name: 2013-01-04 00:00:00, dtype: float64 #this line of code returns a specific range df.iloc[2:4, 0:2]

AB

2013-01-031.5242271.863575

2013-01-040.918203-0.158800

#the following lines of code return specific columns and rows using lists with row and columns indices

df.iloc[[1,5],[0,2]]

AC

2013-01-02-1.0702150.604572

2013-01-060.222260-0.045748

Using Pre-Defined Conditions

You can also perform Boolean indexing using different values present in the columns. This is the only way you can filter the data set based on some predefined conditions.

df[df.A > 1]

ABCD

2013-01-011.030816-1.2769890.837720-1.490111

2013-01-031.5242271.8635751.2913781.300696

#we will now copy the data

df2 = df.copy()

df2['E']=['one', 'one','two','three','four','three']

df2

ABCDE

2013-01-011.030816-1.2769890.837720-1.490111one

2013-01-02-1.070215-0.2091290.604572-1.743058one

2013-01-031.5242271.8635751.2913781.300696two

2013-01-040.918203-0.158800-0.964063-1.990779three

2013-01-050.0897310.114854-0.5858150.298772four

2013-01-060.2222600.435183-0.0457480.049898three #this line selects rows based on the values in columns df2[df2['E'].isin(['two','four'])] ABCDE

2013-01-031.5242271.8635751.2913781.300696two

2013-01-050.0897310.114854-0.5858150.298772four #this line selects those rows without two and four df2[~df2['E'].isin(['two','four'])] ABCDE

2013-01-011.030816-1.2769890.837720-1.490111one

2013-01-02-1.070215-0.2091290.604572-1.743058one

2013-01-040.918203-0.158800-0.964063-1.990779three

2013-01-060.2222600.435183-0.0457480.049898three

The query method in the pandas library allows you to select the required columns in the data set. You only need one criterion to perform this function.

#This is a list of columns where the value A is greater than C

df.query('A > C')

ABCD

2013-01-011.030816-1.2769890.837720-1.490111

2013-01-031.5242271.8635751.2913781.300696

2013-01-040.918203-0.158800-0.964063-1.990779

2013-01-050.0897310.114854-0.5858150.298772

2013-01-060.2222600.435183-0.0457480.049898

#using OR condition

df.query('A < B | C > A')

ABCD

2013-01-02-1.070215-0.2091290.604572-1.743058

2013-01-031.5242271.8635751.2913781.300696

2013-01-050.0897310.114854-0.5858150.298772

2013-01-060.2222600.435183-0.0457480.049898

Pivot tables can be used to customize the information in an easily readable manner. You can use pivot tables to learn more about the data set. Some people use Excel to build and develop pivot tables to better analyze and understand the data.

#create a data frame

data = pd.DataFrame({'group': ['a', 'a', 'a', 'b','b', 'b', 'c', 'c','c'],

'ounces': [4, 3, 12, 6, 7.5, 8, 3, 5, 6]})

data

groupounces

0a4.0

1a3.0

2a12.0

3b6.0

4b7.5

5b8.0

6c3.0

7c5.0

8c6.0

The following statements calculate he mean of the groups

data.pivot_table(values='ounces',index='group',aggfunc=np.mean)

group

6.333333 b 7.166667 c 4.666667

Name: ounces, dtype: float64 #calculate count by each group

data.pivot_table(values='ounces',index='group',aggfunc='count') group

3

3 c 3

Name: ounces, dtype: int64

Let us now look at a real-world example and see how the Pandas and NumPy libraries can be used.

Chapter Seven: Exploring the Data Set

In this chapter, we will use the different functions listed above to obtain information about adults. To work with the same data set, you can download it from the following website: https://s3-ap-southeast-1.amazonaws.com/he-public-data/datafiles19cdaf8.zip . This data has data that is classified using binary conditions. This example aims to calculate the salary of a person using specific variables in the data set.

#load the data

train = pd.read_csv("~/Adult/train.csv")

test = pd.read_csv("~/Adult/test.csv")

#check data set

train.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 32561 entries, 0 to 32560

Data columns (total 15 columns):

age 32561 non-null int64

workclass 30725 non-null object

fnlwgt 32561 non-null int64

education 32561 non-null object

education.num 32561 non-null int64

marital.status 32561 non-null object

occupation 30718 non-null object

relationship 32561 non-null object

race 32561 non-null object

sex 32561 non-null object

capital.gain 32561 non-null int64

capital.loss 32561 non-null int64

hours.per.week 32561 non-null int64

native.country 31978 non-null object

target 32561 non-null object

dtypes: int64(6), object(9)

memory usage: 3.7+ MB

We are going to split the data into two parts – the training and testing data sets. The training data set we will use has 32561 rows and 15 columns. Of these fifteen columns, only six of them have integer values, while the other have character or object data. You can check the values and records present in the testing data set in a similar manner. The number of columns and rows in the data set can be defined using the lines of code written below:

```
print ("The train data has",train.shape)

print ("The test data has",test.shape)
```

('The train data has', (32561, 15))

('The test data has', (16281, 15))

```
#Let have a glimpse of the data set

train.head()
```

ageworkclassfnlwgteducationeducation.nummarital.statusoccupationrelationshipracesexcapital.gaincapit

al.losshours.per.weeknative.countrytarget

039State-gov77516Bachelors13Never-marriedAdm-clericalNot-in-familyWhiteMale2174040United-States<=50K

150Self-emp-not-inc83311Bachelors13Married-civ-spouseExec-managerialHusbandWhiteMale0013United-States<=50K

238Private215646HS-grad9DivorcedHandlers-cleanersNot-in-familyWhiteMale0040United-States<=50K

353Private23472111th7Married-civ-spouseHandlers-cleanersHusbandBlackMale0040United-States<=50K

428Private338409Bachelors13Married-civ-spouseProf-specialtyWifeBlackFemale0040Cuba<=50K Let us verify if the data set has any missing values. nans = train.shape[0] - train.dropna().shape[0]

print ("%d rows have missing values in the train data" %nans)

nand = test.shape[0] - test.dropna().shape[0]

print ("%d rows have missing values in the test data" %nand)

When you run the above lines of code, you will see there are 2399 rows in the training data set with missing values. There are 1221 rows in the test data set with missing values. Let us now look for the columns with missing values.

#only 3 columns have missing values

train.isnull().sum()

age0

workclass1836

fnlwgt0

education0

education.num0

marital.status0

occupation1843

relationship0

race0

sex0

capital.gain0

capital.loss0

hours.per.week0

native.country583

target0

dtype: int64

When you use these character variables, you can count the different values in the data set. These values will be unique.

```
cat = train.select_dtypes(include=['O'])

cat.apply(pd.Series.nunique)
```

workclass8

education16

marital.status7

occupation14

relationship6

race5

sex2

native.country41

target2

dtype: int64

From the previous section, we learned how you can work with missing values in the data set. Let us now see how you can substitute the missing values from different values.

```
#Education

train.workclass.value_counts(sort=True)

train.workclass.fillna('Private',inplace=True)

#Occupation

train.occupation.value_counts(sort=True)

train.occupation.fillna('Prof-specialty',inplace=True)

#Native Country

train['native.country'].value_counts(sort=True)

train['native.country'].fillna('United-States',inplace=True)

train.isnull().sum()
```

age0

workclass0

fnlwgt0

education0

education.num0

marital.status0

occupation0

relationship0

race0

sex0

capital.gain0

capital.loss0

hours.per.week0

native.country0

target0

dtype: int64

We will look at the target variables to see if there are any issues in the data set.

#check proportion of target variable

train.target.value_counts()/train.shape[0]

<=50K 0.75919

>50K 0.24081

Name: target, dtype: float64

pd.crosstab(train.education, train.target,margins=True)/train.shape[0]

target<=50K>50KAll

education

10th0.0267500.0019040.028654

11th0.0342430.0018430.036086

12th0.0122850.0010130.013298

1st-4th0.0049750.0001840.005160

5th-6th0.0097360.0004910.010227

7th-8th0.0186110.0012280.019840

9th0.0149570.0008290.015786

Assoc-acdm0.0246310.0081390.032769

Assoc-voc0.0313570.0110870.042443

Bachelors0.0962500.0682100.164461

Doctorate0.0032860.0093980.012684

HS-grad0.2710600.0514420.322502

Masters0.0234640.0294520.052916

Preschool0.0015660.0000000.001566

Prof-school0.0046990.0129910.017690

Some-college0.1813210.0425970.223918

All0.7591900.2408101.000000

From the data above, you will see that 75% of the people in the data set have a salary of above $50,000 and 27% of these individuals have passed high school. This is accurate since people with lower levels of education will not make a salary over $50,000.

#The following statements are to load sklearn to encode the variables using an object type

from sklearn import preprocessing

for x in train.columns:

if train[x].dtype == 'object':

lbl = preprocessing.LabelEncoder()

lbl.fit(list(train[x].values))

train[x] = lbl.transform(list(train[x].values))

We will now look at the different changes that have been applied to the data set.

train.head()

ageworkclassfnlwgteducationeducation.nummarital.statusoccupationrelationshipracesexcapital.gaincapit al.losshours.per.weeknative.countrytarget

039677516913401412174040380

150583311913230410013380

238321564611905141004038 0

353323472117250210040380

428333840991329520004040

If you pay attention to the output of the above code, you will see that the variables have been converted into numeric data types.

#<50K = 0 and >50K = 1

train.target.value_counts()

24720

1 7841

Name: target, dtype: int64 Building a Random Forest Model

We are now going to test the accuracy of the model using a random forest model.

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.cross_validation import cross_val_score

from sklearn.metrics import accuracy_score

y = train['target'] del train['target'] X = train

X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_state=1,stratify=y) #train the RF

classifier

clf = RandomForestClassifier(n_estimators = 500, max_depth = 6) clf.fit(X_train,y_train)

RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini', max_depth=6, max_features='auto', max_leaf_nodes=None, min_impurity_split=1e-07, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=500, n_jobs=1, oob_score=False, random_state=None, verbose=0, warm_start=False)

clf.predict(X_test)

We will now check the accuracy of the model on the basis of some predictions that are want to make about the data set.

#make prediction and check model's accuracy

prediction = clf.predict(X_test)

acc = accuracy_score(np.array(y_test),prediction)

print ('The accuracy of Random Forest is {}'.format(acc))

The accuracy of Random Forest is 0.85198075545.

The algorithm we used above will give you results with an 85% accuracy. If you want to improve the model’s accuracy, you need to tweak the program and functions used. We are dividing the data set into two parts – the testing and training data set. We use the training data set to enable the model to predict the required outcomes. We then run the testing data set through the model to see if it performs well and predicts an output similar to the training data set.

Chapter Eight: How to Summarize Data with Python

For the purpose of this chapter, we will use the data set for supermarket sales. You can download this data using the following link: https://www.kaggle.com/aungpyaeap/supermarket-sales/data .

Download this information and save it on your local drive for use. You can import the information onto python using the following lines of code:

```
#import library

import pandas as pd

#import file

ss = pd.read_csv('supermarket_sales.csv')

#preview data

ss.head()
```

Obtain Information About the Data Set

You can use the following functions to obtain information about the data set.

Info()

This function gives you a concise summary of the data frame extracted from the data set you have uploaded. This function can be used if you use the Pandas library. This function, among other functions, can be used to perform data cleaning operations. The function provides all the information you need about the data set, such as record counts, columns, data types, memory usage, names of columns and rows, and the index range. When you look at the summary provided for the above data set, you can obtain the following information:

- The record count is 1000
- There are 17 columns in the data set
- You can update the column names if you want to remove any white paces

- Identify the data types
- The date and time columns in the data set use the object type dtype. You need to update these to match the required formats
- The data set does not have any missing values

Describe()

This function gives you the descriptive statistics which will provide you the information you need about the distribution the data set follows. This function does not look at the NaN values, and it can be used on a specific series of data or data frames.

The results will only be for the numeric data types, but you can use the function include if you want to obtain some statistics for every non-numeric data type in the data set. If you only want to see the results for specific data frames, you can use the exclude function. The function, by default, will return the percentile values. The percentiles extracted are in the 25^{th} , $50^{th,}$ and 75^{th} range. If you want to update the different percentiles obtained, you can add the percentiles using the percentiles function. The values used in this function should lie between the values 0 and 1. This function will apply to both series and data frames.

Value_counts()

This function returns unique values for the series specified in the function. This function also excludes NaN values. This function's results are in descending order, but you can always change the way the data is presented using the function **ascending = true** . If you want to obtain the frequency of every value in the data set, you need to set the function **normalize = true** .

nunique()

You can use this function if you count the number of distinct observations. You can use it on series or data frames. This function, by default, excludes the NaN values. If you want to use the NaN values in the data set, you need to use the parameter **dropna = false** . Since there are no missing values in the data set, let us create a data frame to understand how this function works on data sets with NaN values.

```
d = {'A': [1, 2, None], 'B': [3, 4, 2], 'C': [3, None, None]}
df = pd.DataFrame(d)
df
```

When you run the dropna function, you can extract the parameters without the NaN values.

Sun()

This function is used to obtain the sum of all the values of the axis requested in the function. You can use this function on both series and data frames. If you want to apply this function only to numeric data types, you can use the parameter **numeric_only = True** . You can also use the min_count parameter if you want to set a condition to let the compiler know when you can apply the function. We will create another data frame, which you can use to demonstrate how this parameter works.

```
d = {'A': [1, 2, None, 5, 8], 'B': [3, 4, 2, 4, 5],
     'C': [3, None, None, 3, 4]}
df = pd.DataFrame(d)
df
```

count()

This function returns the number of null or non-null observations, and you can apply this function on both series and data frames. If you want to obtain the information only on numeric data types, you should use the parameter **numeric_only = True** .

Minimum, Maximum, Mean, and Median

The following are the functions you can use to extract the minimum, maximum, mean, and median values.

- Min()
- Max()
- Mean()

- Median()

You can apply these functions to both series and data frames.

Chapter Nine: Steps to Build Data Analysis Models in Python

You have come quite far now and have learned the basics of data mining and data analysis. We will now look at how you can use Python to develop a predictive analytics model. We will see how you can divide the data into a training and testing data set and use it to develop and measure the model's performance. When you build any model, you need to spend enough time understanding the business need and developing the hypothesis based on that need. To do this, you need to understand the domain. When you do this, you can identify the problems and queries and map the data to them. This is the only way you can design the business solution to the problem or query. The following are some reasons why you need to do this:

- You can spend enough time to learn more about the data analysis process
- You will not have any other perspective or thoughts which could lead to a bias

This ensures that the hypothesis you develop will not have any bias.

You need to spend sufficient time to complete this stage, and therefore, you need to make this your standard practice. If you perform this step carefully, you will not have to work on multiple iterations over the course of the development of the model. Let us look at the different steps followed in this process. We will look at the time you spend during each stage.

- The first thing you need to do is perform a descriptive analysis of the data you have collected. This will take a majority of your time
- The second step includes the cleaning and treating of data. In this step, you look for any missing values and outliers in the data set. You may spend close to 40% of your time doing this, so make sure you obtain the right data set
- When you obtain the required data set, you need to start the data modeling process. Since you have cleaned and processed the data, you can use different algorithms to develop the model
- The last step is to estimate the model and its performance

This is the time you spend when you develop the model for the first time. You may spend a lot of time to work through the model and make the necessary updates. We will go through each step to see what the step entails and how much time you should spend on it.

Step One: Descriptive Analysis and Data Exploration

If you are starting off as a data scientist or analyst, this may take you some time. When you get the hang of the process, you can use the different techniques to automate numerous operations. It will take you a long time to prepare the data you collected before analyzing or interpreting the results. It is because of this people use different libraries in Python to automate these processes. We looked at some of these processes in the previous chapters in the book.

Some people use advanced deep learning and machine learning tools to improve the task they are performing. These tools also reduce the time you take to perform the task. Do not use complex concepts, such as feature engineering, if you want to reduce the time you spend performing this task. The objective of this step is to identify any missing values or features in the data set. You can perform the following steps to achieve this objective:

- Identify the ID, target features, and inputs in the data set
- Differentiate between categorical and numerical features
- Identify the columns with missing values

Step Two: Remove Missing Values in the Data Set

There are numerous ways to work with missing values in a data set. We looked at these in great detail in the previous chapters. The following are some smart and quick techniques, and these will help you develop effective models.

- You can create a dummy value or flag to replace any missing values in the data set. It is best to use this technique only when there is minimal or no information about the missing values in the data set
- You can use measures of central tendency, such as mean, median, and mode, to replace any missing values in the data set. The

mean and median measures perform extremely well. Experts use the mean to replace any missing values regardless of the data they are dealing with. Having said that, if the data follows a skewed distribution, you should use the median.

- It is best to replace missing values in the data set using different variables in the data set. If you know the type of data you are using, you can create new labels that can be used to replace any missing values in the data set using categorical variables. You can also use a frequency mix to replace any missing values in the data set using a value with higher frequencies.

It is best to reduce the time you spend on this activity by five minutes. Therefore, you need to use simple methods.

Step Three: Data Modeling

You can use different techniques, such as GBM or random forest modeling, to develop the data model. You need to select the model you want to use depending on the problem or query you want to address. These models are the best way to develop or create benchmark solutions. Most data scientists or analysts use any of the above methods to develop their models. You can continue to use the same techniques to develop the model.

Step Four: Estimate the Model's Performance

Once you develop a model and write the code to build it, you need to use different methods to estimate the performance of the model. You should first divide the data set into a training and testing data set. Using the testing data set, you can determine the accuracy of the model. It is best to split the data into the training and testing data set using the 70:30 ratio. This is extremely easy to do, and we will look at how you can do this in the next chapter.

Chapter Ten: How to Build the Model

We will put the techniques we have discussed above into action in this chapter. We will build the required model using Python.

Step One: Import Libraries Required to Develop the Model and Perform the Analysis. Read the Information and Split the Data into Training and Testing Data Sets.

```
import pandas as pd

import numpy as np

from sklearn.preprocessing import LabelEncoder

import random

from sklearn.ensemble import RandomForestClassifier

from sklearn.ensemble import GradientBoostingClassifier

train=pd.read_csv('C:/Users/Analytics
Vidhya/Desktop/challenge/Train.csv')

test=pd.read_csv('C:/Users/Analytics Vidhya/Desktop/challenge/Test.csv')

train['Type']='Train' #These lines create a flag for the test and train data sets

test['Type']='Test'

fullData = pd.concat([train,test],axis=0) #We are combining the training and
testing data sets
```

Step Two: Summarize the Data Set into Columns

```
fullData.columns #This gives you the column headers as the output

fullData.head(10) #This shows the first ten records in your frame

fullData.describe() #This shows you the statistical summary of the data set
```

Step Three: Identification of Variables

In this step, we will identify the following:

- ID Variables
- Target Variables
- Categorical Variables
- Numerical Variables
- Other Variables

```
ID_col = ['REF_NO']

target_col = ["Account.Status"]

cat_cols = ['children','age_band','status','occupation','occupation_partner','home_status','family_income','self_employed','self_employed_partner','year_last_moved','TVarea','post_code','post_area','gender','region']

num_cols= list(set(list(fullData.columns))-set(cat_cols)-set(ID_col)-set(target_col)-set(data_col))

other_col=['Type'] #Test and Train Data set identifier
```

Step Four: Identify the Missing Values in the Data Set and Create a Flag

fullData.isnull().any()#This function returns the values true and false. The former indicates a missing value while the latter indicates there are no missing values.

```
num_cat_cols = num_cols+cat_cols # Combined numerical and Categorical variables

#For evert variable with a missing value create another variables with VariableName_NA

# Use the numbers 1 and 0 to flag missing values

for var in num_cat_cols:
    if fullData[var].isnull().any()==True:
        fullData[var+'_NA']=fullData[var].isnull()*1
```

Step Five: Remove Missing Values

#We will replace or impute a missing numerical value with the mean

fullData[num_cols] = fullData[num_cols].fillna(fullData[num_cols].mean(),inplace=True)

#We will replace or impute any missing categorical value with -9999

fullData[cat_cols] = fullData[cat_cols].fillna(value = -9999)

Step Six: Create Label Encoders for Categorical Variables

When you perform this step, you can split the data set into training and testing data sets. You can split the training data set further into the training and validation data set.

#We will create a label encoder to identify categorical features

for var in cat_cols:

number = LabelEncoder()

fullData[var] = number.fit_transform(fullData[var].astype('str'))

We convert the target variable since it is categorical.

fullData["Account.Status"] = number.fit_transform(fullData["Account.Status"].astype('str'))

train=fullData[fullData['Type']=='Train']

test=fullData[fullData['Type']=='Test']

train['is_train'] = np.random.uniform(0, 1, len(train)) <= .75

Train, Validate = train[train['is_train']==True], train[train['is_train']==False]

Step Seven: Remove Missing Values by Substituting the Dummy Variables

We will use the random forest model to predict the type of data.

features=list(set(list(fullData.columns))-set(ID_col)-set(target_col)-set(other_col))

x_train = Train[list(features)].values

y_train = Train["Account.Status"].values

```
x_validate = Validate[list(features)].values

y_validate = Validate["Account.Status"].values

x_test=test[list(features)].values

random.seed(100)

rf = RandomForestClassifier(n_estimators=1000)

rf.fit(x_train, y_train)
```

Step Eight: Check the Model's Performance and Make Predictions Based on the Analysis Results

Now that the model is ready, you can check the performance and make the necessary predictions.

```
status = rf.predict_proba(x_validate)

fpr, tpr, _ = roc_curve(y_validate, status[:,1])

roc_auc = auc(fpr, tpr)

print roc_auc

final_status = rf.predict_proba(x_test)

test["Account.Status"]=final_status[:,1]

test.to_csv('C:/Users/Analytics
Vidhya/Desktop/model_output.csv',columns=['REF_NO','Account.Status'])
```

Chapter Eleven: Data Visualization

It is important to note that people absorb information better when it is represented in a visual format. When you have performed the analysis and obtained the results, you need to find a way to interpret them. This is one of the biggest challenges that people have faced since there is a lot of information, which can be visualized. If the data is not visualized accurately, it can lead to incorrect interpretations that can impact business decisions. Therefore, as a data scientist or analyst, you need to know exactly what you are looking for in the data set and use different visual aids to depict the data.

Unfortunately, people cannot retain information in the form of text for long periods. Our brains can retain large volumes of information if they are in the visual format. Therefore, it is important to represent information in a visual format to ensure business stakeholders know exactly the analysis's objective. You can use different formats, such as pie charts, maps, spreadsheets, pivot tables, etc., to depict the information in an easy-to-understand manner.

You can convey critical concepts and aspects of the data using intuitive, swift, and simple visualization tools and techniques. Visualization also helps analysts work with different forms of data. They can visualize how the data functions in different formats or scenarios that enable them to develop a robust business model. They can also make the necessary adjustments to the data set.

Data visualization is essential for organizations since it reduces the time spent in meeting rooms by at least 30%. When you represent text and numbers in the form of charts and tables, people can easily determine what you are trying to tell them. They may only have questions about how you developed the graph or table. Now that you know the importance of data visualization, let us look at some aspects to keep in mind when you develop visualization for your business.

Know Your Audience

Most people forget to understand that the objective of any visualization is to make sure the viewer understands the most important data visualization

concepts. Therefore, it is important to depict the data in a format that appeals to the users. You need to find a way to tailor the visualization to suit the audience to whom you are presenting the data.

Set Your Goals

When you create the visualization for the data analysis results, you need to ensure the visuals and aids you use show logical insights and narratives relevant to the business problem being discussed. When you create a goal for your campaigns and pursuits, you need to sit with the business stakeholders and explain your goals and objectives to them.

Choosing the Right Charts to Represent Data

You need to choose the right type of chart to represent the data set's data points. This form of visualization plays an important role since it is the only way you can effectively represent the data. You need to keep the purpose of the project, audience, and project in concern. For instance, if the project or query you are working on looks for the business's changes over a few months, using a simple bar graph or line graph to represent the data. These are some of the best ways to represent data visually. The following are some visualization tools and techniques you can use to represent your data set.

Number Charts

Using number charts, you can effectively and efficiently depict information, which is in numeric formats. You can use number charts to depict the number of times a user entered a website, the type of pictures a user likes on Instagram, and the organization's sales.

Maps

One of the advantages of using maps to represent the data graphically is that they are fun to look at. This means the audience will always be engaged when they look at maps. Another advantage of using maps is you can easily represent information in the data set. You can also represent large volumes

of complex information regarding different topics using maps. It is easy to digest the information depicted through maps.

Pie Charts

Pie charts are used as a traditional way to represent information. Most people do not like using pie charts to represent data, and these charts have received negative feedback over the years. Pie charts are still a great way to visualize data in the data set.

Gauge Charts

You can represent data with single data points or values using gauge charts. These charts are the best forms of visual representation that display trends and indicate any hidden correlations in the data set. You can also use them on executive dashboard reports shared with stakeholders.

These are not the only examples of data visualization techniques. There are other complex forms of visualization that you can use to ensure the business and stakeholders understand the analysis performed.

Conclusion

If you are looking to work with large volumes of data, you have come to the right place. This book has all the information you need to know about data mining, analysis, and visualization. Data mining is the process of extracting and cleaning data collected from different sources that will help you interpret and analyze the information to make informed business decisions.

The book also talks about data analysis and its importance. You will also learn more about how you can use Python to manipulate and work with different data sets. The book also leaves you with an example of how you can develop a predictive model using Python. You can use the book's code to help you learn more about how you can develop different models to work with large volumes of data.

If you are keen on working in the field of data analysis or science, you need to learn how to code in Python. Most professionals use this language to develop robust models. You also need to understand the importance of data visualization, and this book throws some light on this topic. It leaves you with some quick data visualization techniques that you can use to let the business know what you are trying to convey through your analysis.

References

Data Cleaning in Python. (2021). Dezyre.com. https://www.dezyre.com/article/data-cleaning-in-python/406

Data Mining Architecture - Javatpoint. (n.d.). Www.Javatpoint.com. https://www.javatpoint.com/data-mining-architecture

Data Mining Techniques - Javatpoint. (n.d.). Www.Javatpoint.com. https://www.javatpoint.com/data-mining-techniques

Data Mining Tutorial - Javatpoint. (n.d.). Www.Javatpoint.com. https://www.javatpoint.com/data-mining

Data Mining Tools - Javatpoint. (n.d.). Www.Javatpoint.com. https://www.javatpoint.com/data-mining-tools

Rodriguez, M. (2020, July 29). How to Summarize Data with Pandas. Medium. https://medium.com/analytics-vidhya/how-to-summarize-data-with-pandas-2c9edffafbaf

What is Data Analysis? Types, Process, Methods, Techniques. (2019, September 7). Guru99.com. https://www.guru99.com/what-is-data-analysis.html

What Is Data Mining? (2019). Oracle.com. https://docs.oracle.com/cd/B28359_01/datamine.111/b28129/process.htm#DMCON002